Phase 4 Report

The goal of the Main Character (Student) is to graduate from SFU. In other words, the Main Character wants to escape from SFU, the setting of the game. Our Main Character will start in a lecture room. The Protagonist must gain knowledge during their time in SFU by reading books and reviewing notes. In order to win the game, which is to escape SFU, the player has to collect all of the Static Rewards (Textbooks) and Bonus Rewards (Cheat Sheets) in order to gain a high enough Score (Course Credits). However, the Main Character (Student) needs to avoid the Moving Enemy (Professor/Raccoons) and Static Enemy (CCTV), which implies that whether or not they graduate will depends on what grade the Professor will give in order to pass the course, and whether they could stay healthy throughout the SFU years without getting attacked by the Raccoons. On another hand, CCTV will symbolize whether the Main Character (Student) will break any school rules or not obey the pledge of academic dishonesty. The goal of the Main Character (Student) is to not get caught by Enemies and collect enough Rewards before reaching the End Goal. The game world is based on SFU, thus, the game will start in the lecture room inside the AQ, once completed, the Main Character (Student) will proceed to outside of AQ, then to the Bus Loop to end the Game.

We came up with several ideas for the game. We decided on SFU Break as the concept excited us. As for the design of the game, we initially planned on building the game using the: MVC Pattern that separates the Game Logic into 3 Components:

1. The Model: Game State and Logic
2. The View: Displays the Game State to the User
3. The Controller: Handles User Input and Updates (Model) and (View) ● The Model:
    - ○ Game World (Board of the Game)
    - ○ Main Character, Enemies, Rewards

● The View:

- ○ The Graphics and User Interface

● The Controller:

- ○ Handles Keyboard Input

However, it seemed too complicated to implement considering our lack of experience with Java. For example, the UML Diagram that we designed in Phase 1 highlighted our initial shortcomings with Java and our structural designing-skills required for the game. We thought we could build the whole game with just 5 main classes, but after we completed the game, we knew it was poorly thought-out, and acknowledged that our initial designs were inadequate. In short, we did not stay faithful to our initial plan and we made sizable changes and improved the design in every phase of the project.

In the end, we implemented the Abstract Factory method as it is easier to implement since we have learned it in class before.

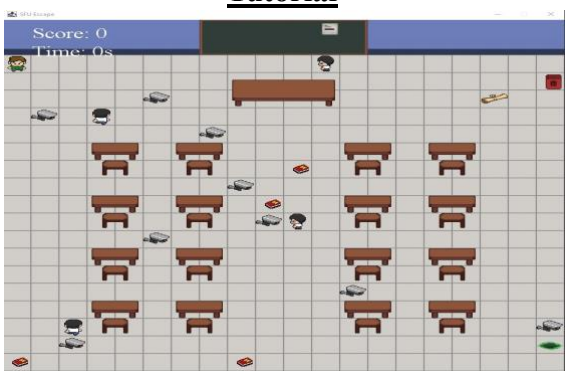
As for the final product, we managed to keep the majority of the plan, especially with the narrative of the game. Having said that, we made large changes in the structure and technology aspects of the game throughout the process.

- We used the Abstract Factory method instead of the MVC method.
- The design of our game objects and characters improved drastically.
- The code of our game has been refactored multiple times as we are improving in our Java proficiency.
- The UML diagram is totally different from what we had initially, since we have almost 20-30 classes.
- The UI also had some modifications as we decided not to include the difficulty level in our game to make it more like a storytelling game.

As far as project and team management is concerned, the two main areas of improvement that we highlighted were communication and experience. We initially believed we had put in an adequate amount of effort to ensure each team member was aware of their relevant goals and objectives. However, we have since learned that we could have emphasized communication even more. We decided early on to adopt an agile software development approach. To improve our communication, we should have appreciated the need to constantly discuss and review each of our perspectives on development. We have learned that this is crucial for managing the chaos inherent in the software development approach. Secondly, we have learned of the importance of experience with every aspect of developing a program. Especially when it came to implementation, we spent a lot more time developing and testing code than we would have if we initially had a higher comprehension of Java.

The main aspect of the software development process that we value more now is the initial design and planning phase.We found that the later we tried to change elements of the program, the more time-consuming it became; as the game was further developed, altering the design meant having to consider and revise an increasing number of code blocks. If we spent more time thinking about and researching for the initial design and planning phase, we probably would have detected and considered a lot of the problems that we ended up facing. This was especially significant when it came to integrating different aspects of the program. When reflecting, it was very apparent that we underestimated the cost of integration. Another aspect of the software development process that we did learn to value more was testing. Planning for testing, especially considering cohesion, was something we should have done more. It became difficult to distinguish the exact issues that were shown in our testing phase. This led to us spending a good amount of time adjusting our code to make it more testable.

Tutorial



As seen in the image above, the game begins with the main character spawning at the top left of a classroom. The room contains physical objects such as desks, trash cans and chairs.

As seen in the image, the primary rewards are Books (+100 points), and the bonus rewards are cheat sheets (+200 points) which spawn at random locations and timings and also disappear after a random amount of time. For punishments, the moving enemies are professors and the static enemies are security cameras. If caught by a professor, the user loses the game. Also, cameras deduct points and if the score drops to a negative number, the user loses the game.

After collecting a minimum of 500 points, the main character can then go through the green portal at the bottom right of the classroom. This takes the user outside, as seen below:

In the new stage, the goal of collecting the rewards stays the same. However, the moving enemies are now Raccoons. They share the same behaviour as the professors. Once the user reaches 1000 points in the second level, they may escape the AQ through the top-middle of the grid (the yellow exit) in order to win the game.

The main character is controlled by the arrow keys on the keyboard. The main character cannot move to the same space as other concrete objects in the game, such as walls, tables, plants, and water. In order to interact with rewards, punishments and exits, the user must and the corresponding object must make contact with each other.